A mathematical description of a line follower robot

Basil Skaria,
S3 Computer Science and Engineering Department (2016-20 Batch),
Muthoot Institute of Technology and Science, Puthencruz.

Abstract : The problem is to describe mathematically a line following robot on a plane represented as a grid [2]. Input to the robot is a set of information like the starting point, end point, the path and the map of obstacles. The path is defined along with a set of undefined obstacles of definite size and the robot is challenged to trace the path from beginning to end.*

Definition 1: *Grid* G is an $N \times M$ matrix, where $N, M \in \mathbb{Z}$.

Definition 2: A *cell* is the smallest entity of a grid. A cell is defined as an ordered pair$\langle x, y \rangle$, where$0 \leq x \leq N, 0 \leq y \leq M$. The dimensions of a cell is the dimension of the robot.

Definition 3: Instruction set for the robot be given by I={N,E,W,S,NE,NW,SE,SW}.

- Upward : $\langle x, y \rangle \rightarrow \langle x, y+1 \rangle$ Notation : N
- Downward : $\langle x, y \rangle \rightarrow \langle x, y-1 \rangle$ Notation : S
- Right : $\langle x, y \rangle \rightarrow \langle x+1, y \rangle$ Notation : E
- Left : $\langle x, y \rangle \rightarrow \langle x-1, y \rangle$ Notation : W
- Up-right : $\langle x, y \rangle \rightarrow \langle x+1, y+1 \rangle$ Notation : NE
- Up-left : $\langle x, y \rangle \rightarrow \langle x-1, y+1 \rangle$ Notation : NW
- Down-right : $\langle x, y \rangle \rightarrow \langle x+1, y-1 \rangle$ Notation : SE
- Down-left : $\langle x, y \rangle \rightarrow \langle x-1, y-1 \rangle$ Notation : SW

$\rightarrow$represents a movement from current position to the next, not the regular if-then symbol.

Definition 4: Cells 'a' and 'b' are *adjacent*, if by any one instruction of set I, a robot can move from 'a' to 'b'.

Definition 5: The sequence of cells is called a *path* represented by $T = t_1 t_2 t_3 ... t_n$, where t_i is a cell and t_i adjacent to t_{i+1} for all i. *Path length* is number of cells that are adjacent to form a path.

Definition 6: A set of cells that a robot cannot traverse in the given map is an *obstacle*. For a path $T = t_1 t_2 t_3 ... t_n$ let $O : T \rightarrow \{TRUE, FALSE\}$, i.e., presence of obstacle maps to TRUE and absence maps to FALSE.

Definition 7: A *valid cell* is one into which a robot can move whereas an *invalid cell* is that a robot cannot move into.

Definition 8: *Initial cell* in a path T is the first cell in it, *terminating cell* is the last cell in the given path T.

Definition 9: A program P is a sequence of instructions. For example, $P = p_1 \cdot p_2 \cdot p_3 \cdot p_4 \cdot p_5 \cdot p_6 \cdot p_7 \cdot p_8$, where $p_i \in I$ and '.' is the concatenation operation.

Theorem 1: Given a grid G and a path T that has no obstacle, there exists an algorithm that can generate a program which moves the robot tracing the path T.

Proof:
For a given path $T = \langle x_1, y_1 \rangle . \langle x_2, y_2 \rangle . \langle x_3, y_3 \rangle \dots \langle x_k, y_k \rangle$, and by definition of path $\langle x_i, y_i \rangle$ and $\langle x_{i+1}, y_{i+1} \rangle$are adjacent and $O\colon T \to \{FALSE\}$.

There exist an $i \in I$ that can take the robot from $\langle x_c, y_c \rangle$ to $\langle x_{c+1}, y_{c+1} \rangle$where i is an instruction, hence we can define a function that maps pairs of cells $\langle x_c, y_c \rangle$and $\langle x_{c+1}, y_{c+1} \rangle$to I. Therefore any pair of cells in T be mapped to I. Then T can be mapped to a sequence of instructions, that will take the robot from initial cell in T ($\langle x_1, y_1 \rangle$), to the terminating cell in T ($\langle x_k, y_k \rangle$). The program executes $k - 1$ number of times.

Results :
The time complexity of the algorithm generated will be as follows:
1. Best case: O(K), where K is the path length.
2. Average case: O(N*M)
3. Worst case: O(N*M), where N and M are the dimensions of the grid G.

The O() represents the asymptotic notation, which determines the lower bound of a given algorithm.

Theorem 2: Given a grid G and a path T that has obstacles only on the path, then there exists an algorithm that can generate a program which moves the robot tracing the valid cells of path T.

Proof: Left for the reader to pursue.

<u>References</u> :
[1] Vladimir J. Lumelsky and Alexander A. Stepanov, Path-Planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape. Algorithmica 9 1987 Springer-Verlag New York Inc.
[2] Girish Balakrishnan, On Building Mathematical Models, 6 Dec 2017, The Blog, Vol. 1, Issue 8.